

## Sheet (2) Solution

Problem #1:

m-n fetches are necessary to make the PC equal to m if the PC is initially equal to n, and there are no branch instructions between n and m.

PC points to the next instruction that is going to be fetched and executed, and IR holds the instruction that is currently being executed. So, when the PC points to n+k, the Instruction Register (IR) holds the instruction at the memory location n+k-1.

Problem #2:

1. The value of the PC is transferred to the Memory Address Register (MAR).
2. Memory\_read control signal is activated.
3. The contents of the memory location pointed to by MAR are transferred to Memory Data Register (MDR).
4. The contents of MDR are transferred to IR.
5. The PC is incremented and points to 1016.

Problem 2.10 from Hamacher

(a)

	MOVE #AVEC, R1	1
	MOVE #BVEC, R2	1
	LOAD N, R3	2
	CLEAR R0	1
LOOP	LOAD (R1)+, R4	2
	LOAD (R2)+, R5	2
	MULTIPLY R4, R5	1
	ADD R5, R0	1
	DECREMENT R3	1
	BRANCH>0 LOOP	1
	STORE R0, DOTPROD	2

Problem 2.13 from Hamacher

(a) EA = 1220

(b) Immediate operand, part of the instruction.

(c) EA = 5830

(d) EA = 4599

(e) EA = 1200

Note: If you assumed that the word length of the processor is 4, and hence your answer to (d) is 4596, that would be acceptable.

2.17 from Hamacher

- (a) MOVE (R5)+, R0  
ADD (R5)+, R0

MOVE R0, -(R5)

(b) MOVE 16(R5), R3

(c) ADD #40, R5

Note: I have assumed the processor word length to be 4. If you assumed any other word length, and answered the question accordingly, that would be considered as well.

Problem 3.29 from Hamacher

(a) They both leave the 17<sup>th</sup> negative word in RSLT.

(b) Both programs scan through the list to find the 17<sup>th</sup> negative number in the list. So, the total number of memory accesses is:  $9 + 7 \times 17 + P \times 3$ .